REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE *(DD-MM-YYYY)*

2. REPORT TYPE
Final

3. DATES COVERED *(From - To)*

4. TITLE AND SUBTITLE
DETECTING NATURAL VERSUS UNNATURAL DNA

5a. CONTRACT NUMBER
FA9550-06-1-0478

5b. GRANT NUMBER
AFOSR

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Dr. Carla Brodley

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Tufts University
Medford, MA

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR
875 Randolph
Arlington VA 22203

10. SPONSOR/MONITOR'S ACRONYM(S)
AFOSR

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Dist A

13. SUPPLEMENTARY NOTES

14. ABSTRACT
We have developed computational pipeline that compares a target genome with related genomes and find regions that have been potentially engineered. Our pipeline compares the target genome with related genomes and finds “unique” genes that have no homologs. These “unique genes” can then be tested for other criteria like DNA composiiton to narrow down the list list of potential engineered genes.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:

a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**

17. LIMITATION OF ABSTRACT

18. NUMBER OF PAGES

19a. NAME OF RESPONSIBLE PERSON

19b. TELEPHONE NUMBER *(Include area code)*

Reset

INSTRUCTIONS FOR COMPLETING SF 298

1. REPORT DATE. Full publication date, including day, month, if available. Must cite at least the year and be Year 2000 compliant, e.g. 30-06-1998; xx-06-1998; xx-xx-1998.

2. REPORT TYPE. State the type of report, such as final, technical, interim, memorandum, master's thesis, progress, quarterly, research, special, group study, etc.

3. DATES COVERED. Indicate the time during which the work was performed and the report was written, e.g., Jun 1997 - Jun 1998; 1-10 Jun 1996; May - Nov 1998; Nov 1998.

4. TITLE. Enter title and subtitle with volume number and part number, if applicable. On classified documents, enter the title classification in parentheses.

5a. CONTRACT NUMBER. Enter all contract numbers as they appear in the report, e.g. F33615-86-C-5169.

5b. GRANT NUMBER. Enter all grant numbers as they appear in the report, e.g. AFOSR-82-1234.

5c. PROGRAM ELEMENT NUMBER. Enter all program element numbers as they appear in the report, e.g. 61101A.

5d. PROJECT NUMBER. Enter all project numbers as they appear in the report, e.g. 1F665702D1257; ILIR.

5e. TASK NUMBER. Enter all task numbers as they appear in the report, e.g. 05; RF0330201; T4112.

5f. WORK UNIT NUMBER. Enter all work unit numbers as they appear in the report, e.g. 001; AFAPL30480105.

6. AUTHOR(S). Enter name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. The form of entry is the last name, first name, middle initial, and additional qualifiers separated by commas, e.g. Smith, Richard, J, Jr.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES). Self-explanatory.

8. PERFORMING ORGANIZATION REPORT NUMBER. Enter all unique alphanumeric report numbers assigned by the performing organization, e.g. BRL-1234; AFWL-TR-85-4017-Vol-21-PT-2.

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES). Enter the name and address of the organization(s) financially responsible for and monitoring the work.

10. SPONSOR/MONITOR'S ACRONYM(S). Enter, if available, e.g. BRL, ARDEC, NADC.

11. SPONSOR/MONITOR'S REPORT NUMBER(S). Enter report number as assigned by the sponsoring/ monitoring agency, if available, e.g. BRL-TR-829; -215.

12. DISTRIBUTION/AVAILABILITY STATEMENT. Use agency-mandated availability statements to indicate the public availability or distribution limitations of the report. If additional limitations/ restrictions or special markings are indicated, follow agency authorization procedures, e.g. RD/FRD, PROPIN, ITAR, etc. Include copyright information.

13. SUPPLEMENTARY NOTES. Enter information not included elsewhere such as: prepared in cooperation with; translation of; report supersedes; old edition number, etc.

14. ABSTRACT. A brief (approximately 200 words) factual summary of the most significant information.

15. SUBJECT TERMS. Key words or phrases identifying major concepts in the report.

16. SECURITY CLASSIFICATION. Enter security classification in accordance with security classification regulations, e.g. U, C, S, etc. If this form contains classified information, stamp classification level on the top and bottom of this page.

17. LIMITATION OF ABSTRACT. This block must be completed to assign a distribution limitation to the abstract. Enter UU (Unclassified Unlimited) or SAR (Same as Report). An entry in this block is necessary if the abstract is to be limited.

Final Report

Carla Brodley, Lenore Cowen, Donna Slonim
Tufts University

Jonathan Eisen
UC, Davis

October 2007

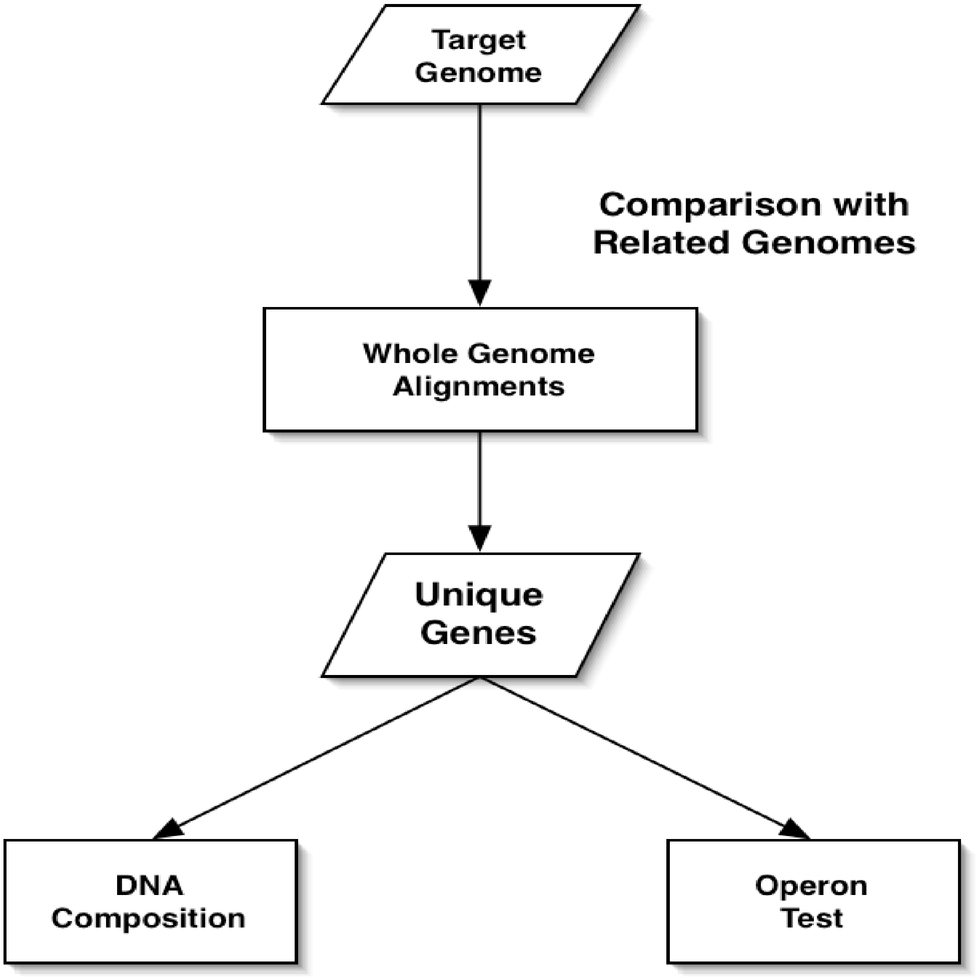
Task 1: Simulating Genetic Engineering

- Goal: To create datasets for testing our detection methods and to share them with the broader scientific community
- No artificially engineered genome is available in GENBANK
- Approach: Design computer programs to simulate tampering of genomes
- We used *E. coli* (K12 strain) genome as the backbone in our simulations
- Create data by tampering with the host genome
  - Insertion of foreign gene into the host genome.
  - Replacement of host genes with orthologous genes from foreign species.
- Varied the difficulty level for detection methods
  - Tampered with genes from species at various distances from *E. coli*.

Task 2: Comparative Genomics Approach

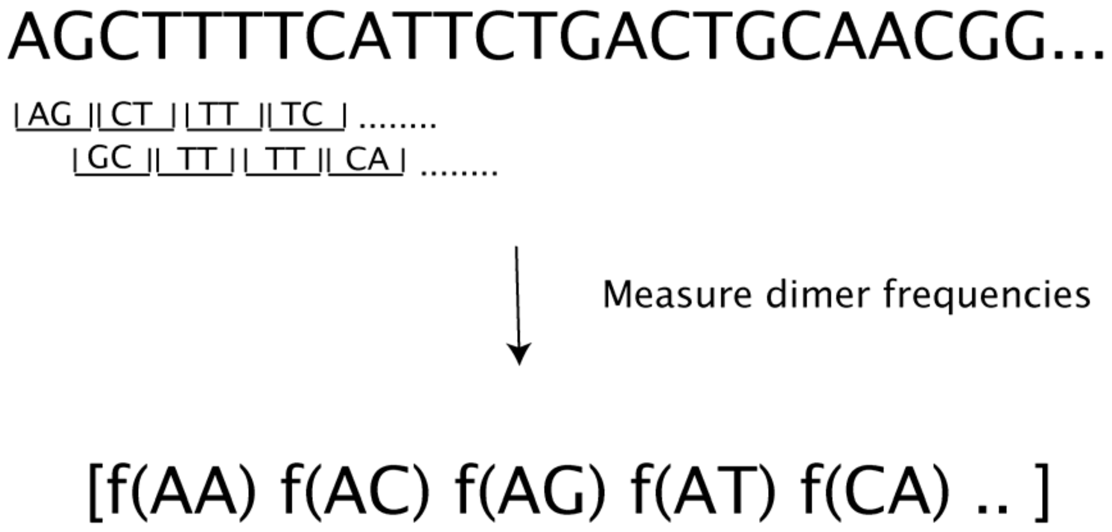
- Goal: Compare a target genome with related isolates and determine whether it has been engineered.
- Why Comparative Genomics?
  - We can tell more about a genome by comparing it with related genomes rather than by looking solely at the genome alone
  - With advent of large scale sequencing, any target genome is likely to have a sequenced relative.
    - More than 480 genomes in GENBANK.

We have developed computational pipeline that compares a target genome with related genomes and find regions that have been potentially engineered. Our pipeline compares the target genome with related genomes and finds “unique” genes that have no homologs. These “unique genes” can then be tested for other criteria like DNA composition to narrow down the list of potential engineered genes.



Task 3: DNA Composition Tests
Can be generalized to K-mers:

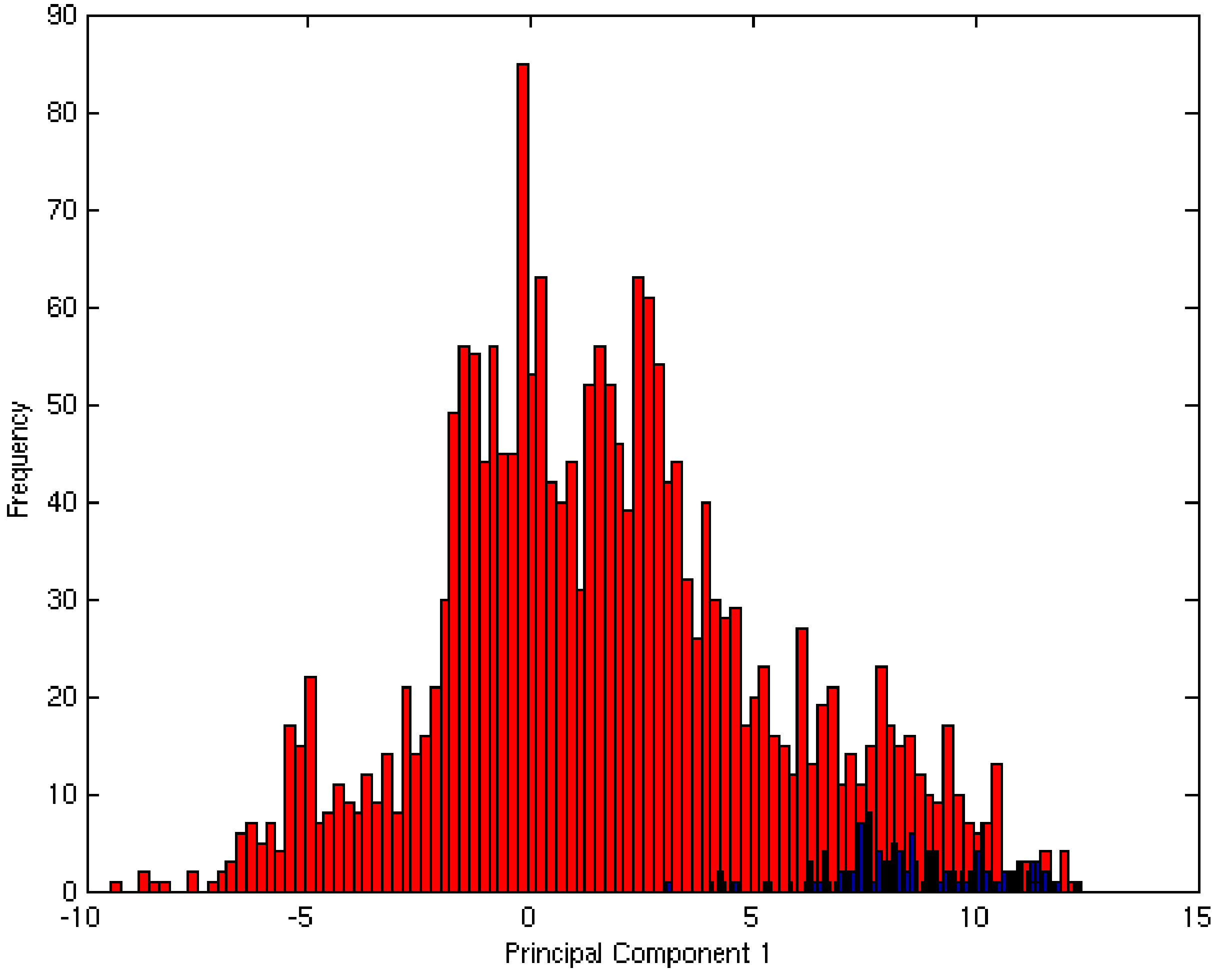
It has been known since the 1960s that different organisms have different “genome signatures”. So engineered DNA will have a different “signature” compared to the signature of host DNA. The most common DNA-composition metric is the Kmer metrics, where we measure the frequency of length K words in a sequence. The figure shows the dimer metric, but we can generalize it for arbitrary K.


AGCTTTTCATTCTGACTGCAACGG...
| AG || CT | | TT || TC | ....
| GC || TT | | TT || CA | ....
Measure dimer frequencies
[f(AA) f(AC) f(AG) f(AT) f(CA) ..]

- Use Hexamer Frequencies.
- Curse of Dimensionality
  - Exponential growth of feature space with higher K.
- Use PCA to deal with the curse of dimensionality
  - First few PCs good for detecting outliers.

We use hexamer frequencies because it can detect biases due to (i) codon bias (ii) restriction enzymes. However, the dimensionality of the data-set increases with the size of K. We deal with this problem by using Principal Component Analysis. The first few principal components are good enough to detect outliers.

Task 4: Comparative Genomics Pipeline



In this figure, we show the first principal component of the hexamer frequency data of 1K fragments taken from the E. coli genome engineered with B. anthracis genes. The host DNA is shown in red, whereas the engineered DNA is shown in red. However this test also detects other anomalous DNA such as the ones due to recent Lateral Gene Transfer.

Operon Test

- Assumption: Adversary inserts a group of contiguous genes or operons [e.g. pathogenic islands].
- Test: Only look for “unique genes” that occur in clusters.
- Caveat: Many toxins can be single genes.

We have also developed an operon test in which we look for clusters of unique genes. This is because many “related genes” like pathogenic islands occurs in clusters in prokaryotes.

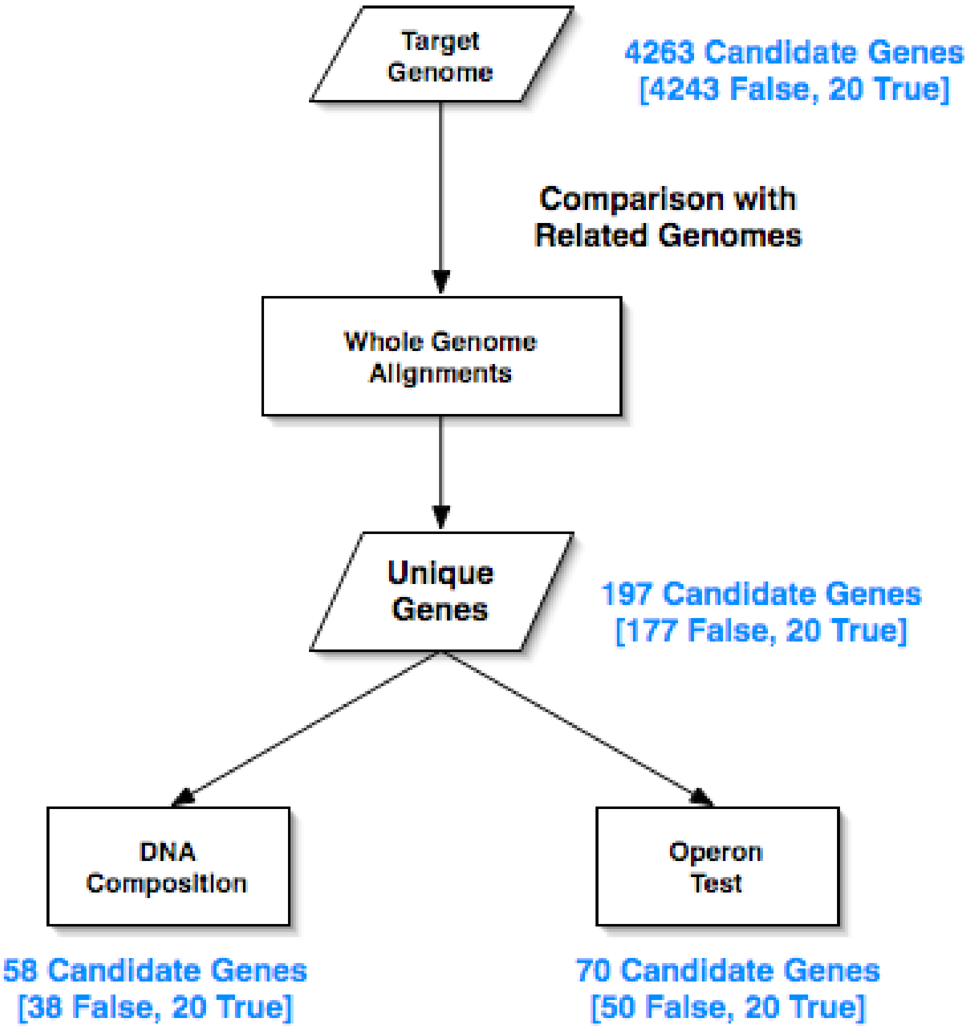
Testing

- Use Simulated Data Sets

  - Insert 20 foreign genes from *Bacillus anthracis* to *Escherichia coli K12* genome.
- Goal: To filter the list of “candidate engineered genes”.

To test our methods, we simulated engineering of 20 Bacillus anthracis genes into E. coli K12 genome. Then we ran our pipeline through our data-set.

Task 4: Comparative Genomics Pipeline

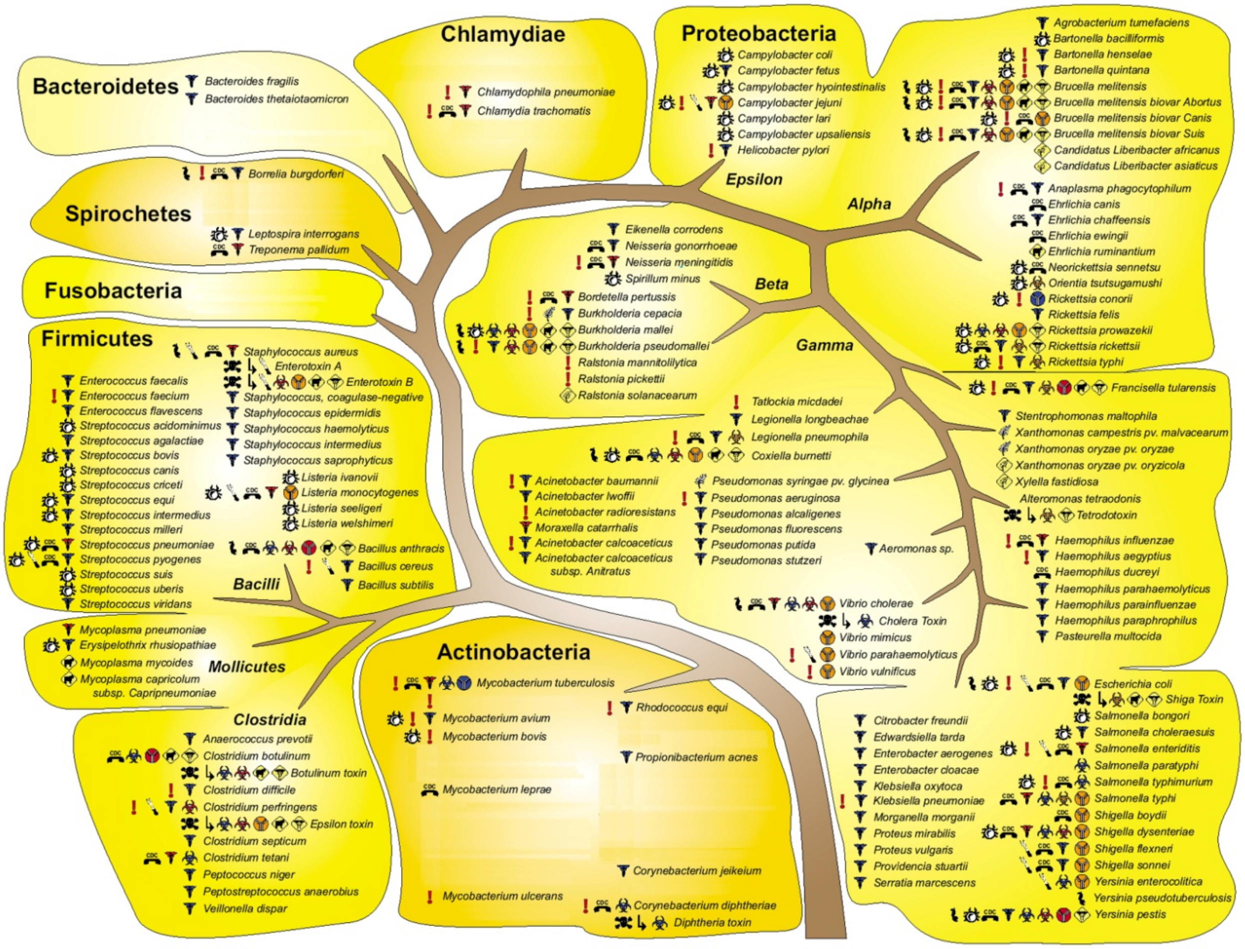


The results of each step of the pipeline are shown in blue. The initial whole genome has 4263 genes of which 20 are “engineered”. After comparing with closely related E. coli genomes, our list was narrowed to 197 genes, including all 20 engineered genes. Additional tests like the DNA composition and operon test further decreased the number of candidate genes.

Task 5: Detect Pathogenic Islands

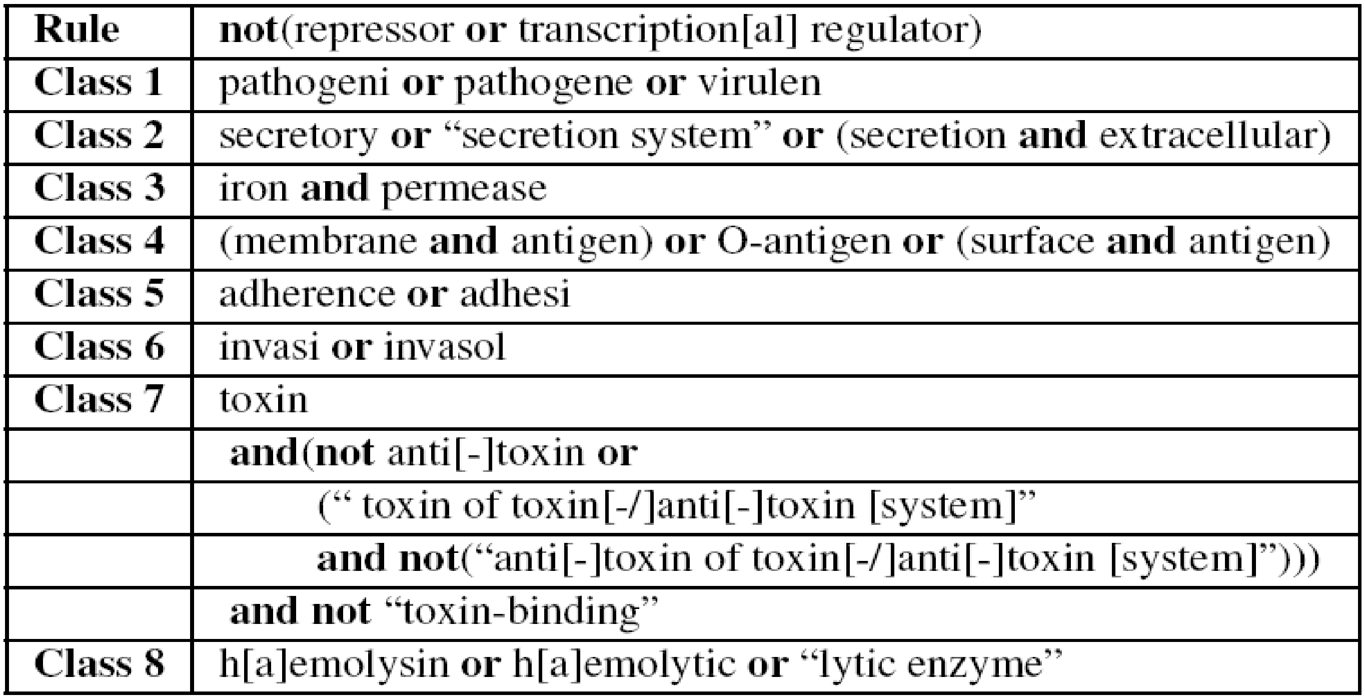
- We seek to answer the following question:

Can we learn functional motifs associated with pathogenicity?



from Ecker et al., "The Microbial Rosetta Stone Database: A compilation of global and emerging infectious microorganisms and bioterrorist threat agents," *BMC Microbiology*, 2005
This is what we know about the "family tree" of pathogenic bacteria to date. We wished to learn signatures of pathogenicity both within and across phyla.

Construction of training set: positive examples



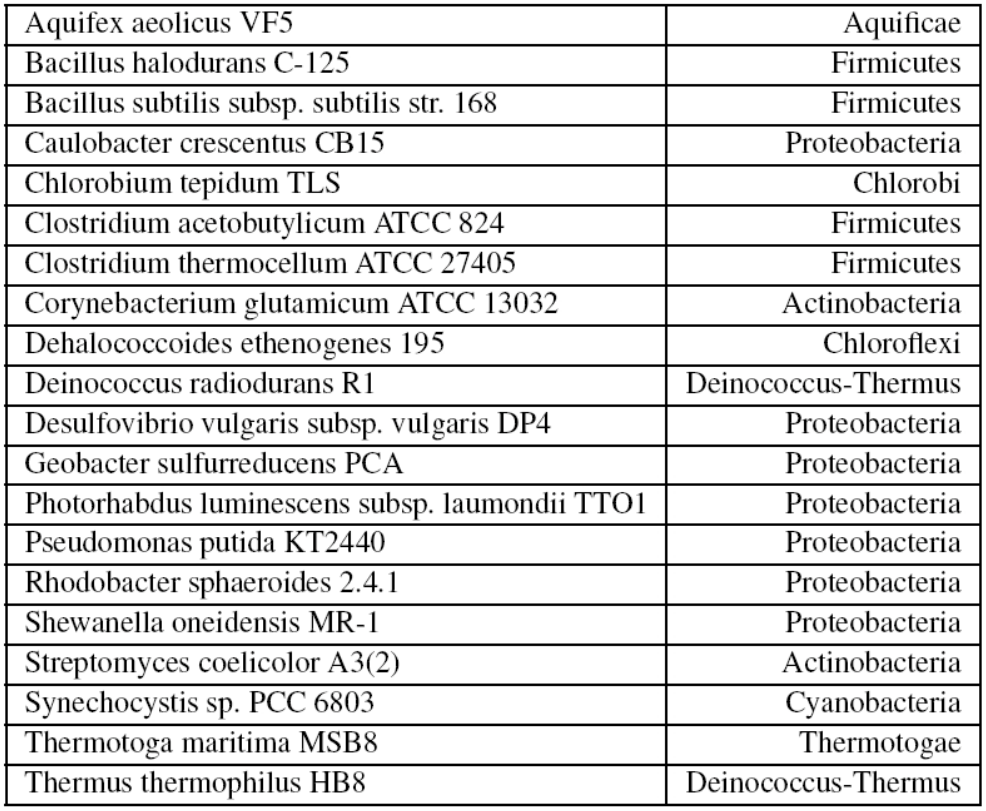
Rule	**not**(repressor **or** transcription[al] regulator)
Class 1	pathogeni **or** pathogene **or** virulen
Class 2	secretory **or** “secretion system” **or** (secretion **and** extracellular)
Class 3	iron **and** permease
Class 4	(membrane **and** antigen) **or** O-antigen **or** (surface **and** antigen)
Class 5	adherence **or** adhesi
Class 6	invasi **or** invasol
Class 7	toxin
	and(**not** anti[-]toxin **or**
	(“ toxin of toxin[-/]anti[-]toxin [system]”
	and not(“anti[-]toxin of toxin[-/]anti[-]toxin [system]”)))
	and not “toxin-binding”
Class 8	h[a]emolysin **or** h[a]emolytic **or** “lytic enzyme”

Our SVM method needs training data to learn signatures of pathogenicity. We accomplish this by mining literature annotation for key words associated with nasty genes. Some of our classes are less specific (class 1-pathogen) and some are more specific (class 8-haemolysin).

Construction of training set: negative examples

Since the function of many genes is unknown, the best way to be sure of having non-pathogenic genes is to take a sampling of random genes from nonpathogenic organisms for negative examples for our training set.

Table 5. Nonpathogenic organisms used to create negative training sets and their phyla.



Aquifex aeolicus VF5	Aquificae
Bacillus halodurans C-125	Firmicutes
Bacillus subtilis subsp. subtilis str. 168	Firmicutes
Caulobacter crescentus CB15	Proteobacteria
Chlorobium tepidum TLS	Chlorobi
Clostridium acetobutylicum ATCC 824	Firmicutes
Clostridium thermocellum ATCC 27405	Firmicutes
Corynebacterium glutamicum ATCC 13032	Actinobacteria
Dehalococcoides ethenogenes 195	Chloroflexi
Deinococcus radiodurans R1	Deinococcus-Thermus
Desulfovibrio vulgaris subsp. vulgaris DP4	Proteobacteria
Geobacter sulfurreducens PCA	Proteobacteria
Photorhabdus luminescens subsp. laumondii TTO1	Proteobacteria
Pseudomonas putida KT2440	Proteobacteria
Rhodobacter sphaeroides 2.4.1	Proteobacteria
Shewanella oneidensis MR-1	Proteobacteria
Streptomyces coelicolor A3(2)	Actinobacteria
Synechocystis sp. PCC 6803	Cyanobacteria
Thermotoga maritima MSB8	Thermotogae
Thermus thermophilus HB8	Deinococcus-Thermus

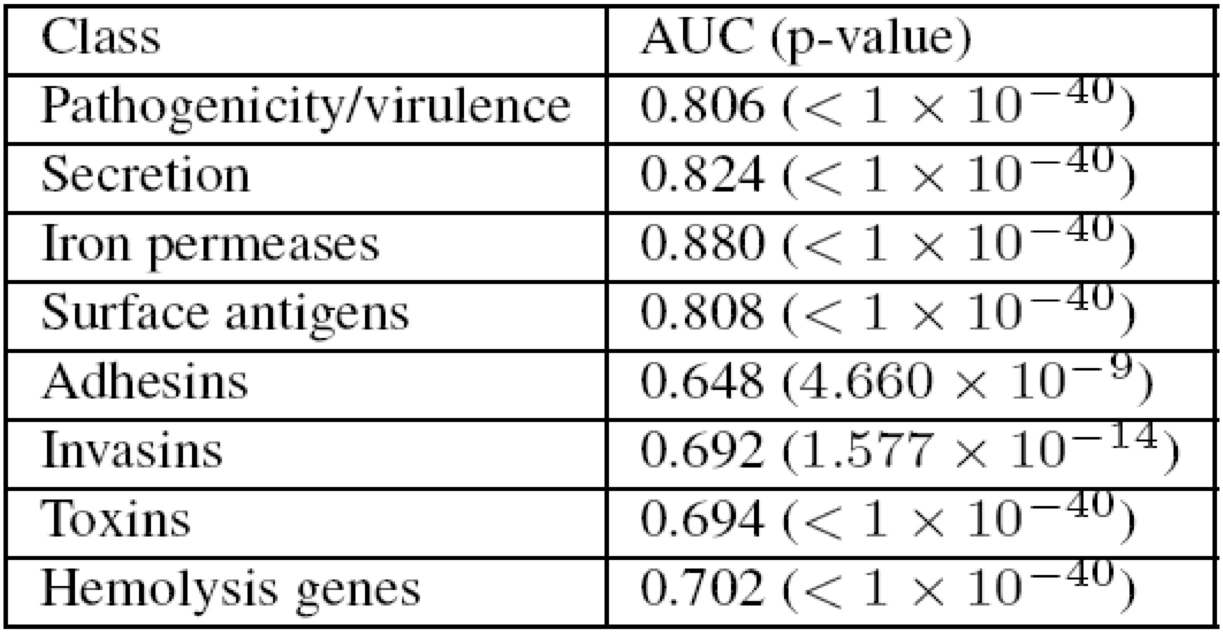
Method

We use support vector machines to learn 8 pathogenic classes of genes, using motifs consisting of very short amino acid strings (simple string kernel method).

SVMs learn classes in a high dimensional feature space. The features that we use are simply frequencies of very short substrings of amino acids (3-4 aa's long; including a wildcard character). It's of independent interest that such short motifs have a functional signal.

Proteobacteria

Here are the results when restricted to Proteobacteria in a 10-fold cross validation study. The "Area Under the Curve" statistic measures the tradeoff of the true positive/true negative rate as a single value– AUC of 1 is perfect classification and 50 percent is random chance. The p-value statistics represent the likelihood of seeing that AUC for that class by chance.

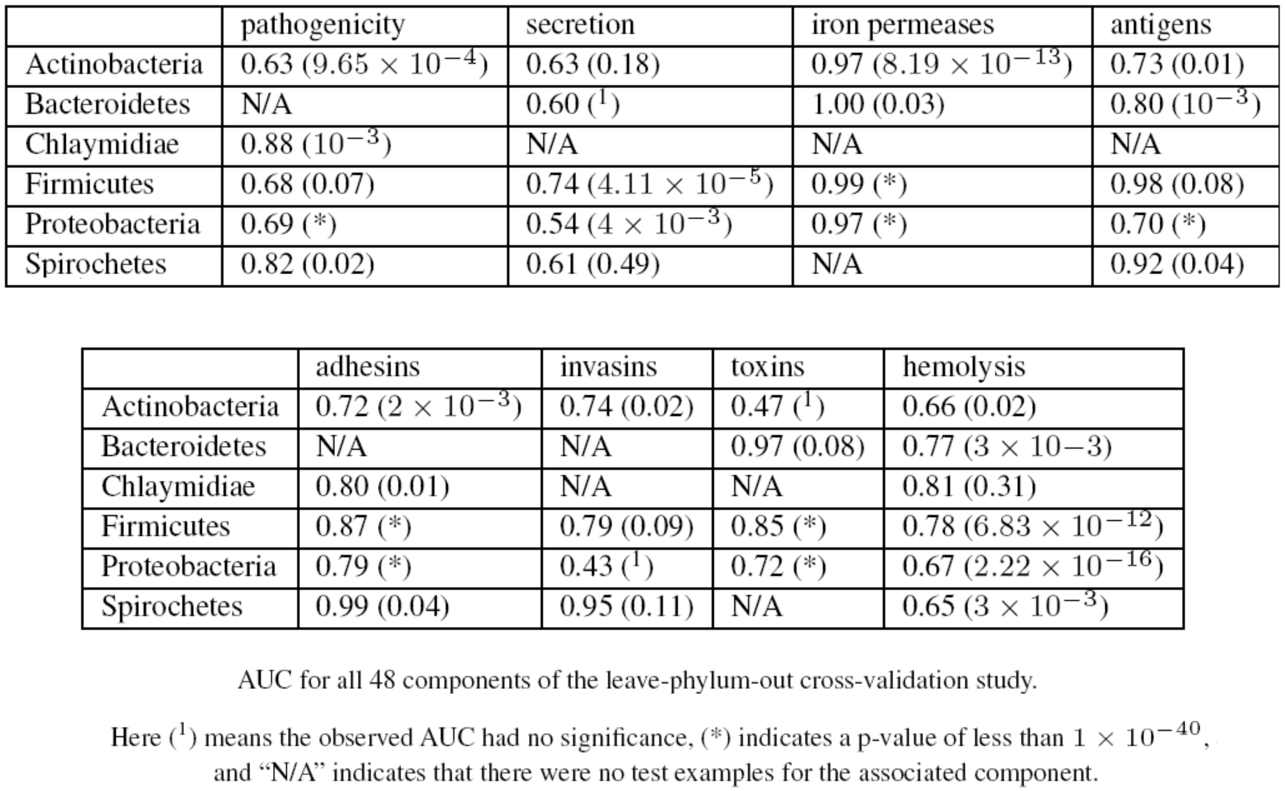


Class	AUC (p-value)
Pathogenicity/virulence	$0.806\ (< 1 \times 10^{-40})$
Secretion	$0.824\ (< 1 \times 10^{-40})$
Iron permeases	$0.880\ (< 1 \times 10^{-40})$
Surface antigens	$0.808\ (< 1 \times 10^{-40})$
Adhesins	$0.648\ (4.660 \times 10^{-9})$
Invasins	$0.692\ (1.577 \times 10^{-14})$
Toxins	$0.694\ (< 1 \times 10^{-40})$
Hemolysis genes	$0.702\ (< 1 \times 10^{-40})$

AUC for all 8 classes in the 10-fold cross-validation study.

Results: Leave-phylum-out cross-validation

Here are the same statistics for a “leave phylum out” cross validation. This is a much harder problem, because we are trying to learn what toxins in Actinobacteria are from, for example, toxins in Proteobacteria. We do surprisingly well, with a couple of exceptions– the classes we do badly with are marked with “1” – in most cases the reason is too few examples; the only exception is we are not doing well at predicting invasins in Proteobacteria when trained only on bacteria outside the Proteobacteria phylum. On closer examination of the data, Proteobacteria contain a set of proteins all homologous to one protein labeled “invasion-like” which is completely dissimilar to examples found outside the phylum. So that’s why we do so poorly on this class.



	pathogenicity	secretion	iron permeases	antigens
Actinobacteria	0.63 (9.65×10^{-4})	0.63 (0.18)	0.97 (8.19×10^{-13})	0.73 (0.01)
Bacteroidetes	N/A	0.60 (^{1})	1.00 (0.03)	0.80 (10^{-3})
Chlaymidiae	0.88 (10^{-3})	N/A	N/A	N/A
Firmicutes	0.68 (0.07)	0.74 (4.11×10^{-5})	0.99 (*)	0.98 (0.08)
Proteobacteria	0.69 (*)	0.54 (4×10^{-3})	0.97 (*)	0.70 (*)
Spirochetes	0.82 (0.02)	0.61 (0.49)	N/A	0.92 (0.04)

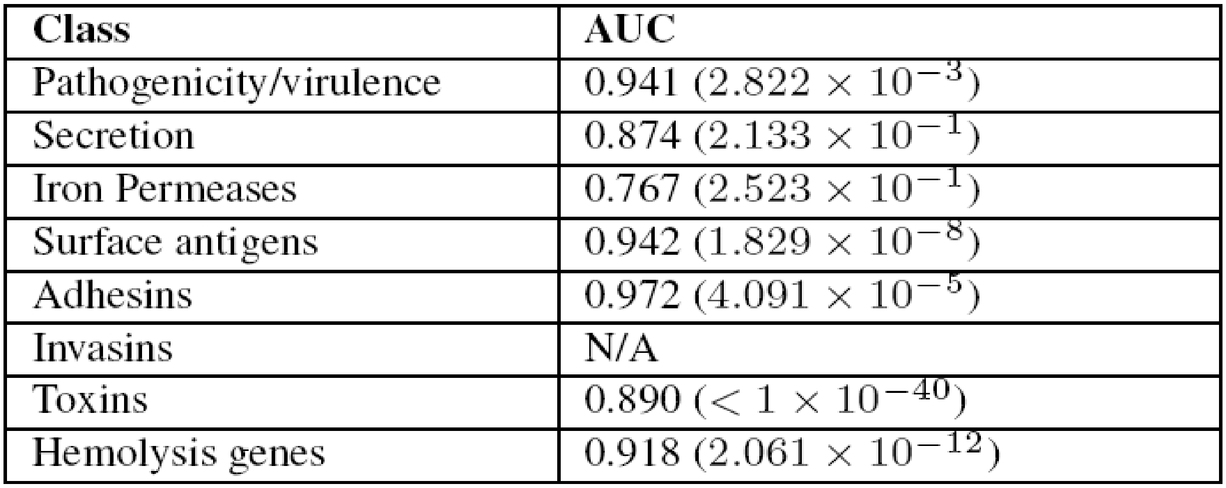
	adhesins	invasins	toxins	hemolysis
Actinobacteria	0.72 (2×10^{-3})	0.74 (0.02)	0.47 (^{1})	0.66 (0.02)
Bacteroidetes	N/A	N/A	0.97 (0.08)	0.77 ($3 \times 10-3$)
Chlaymidiae	0.80 (0.01)	N/A	N/A	0.81 (0.31)
Firmicutes	0.87 (*)	0.79 (0.09)	0.85 (*)	0.78 (6.83×10^{-12})
Proteobacteria	0.79 (*)	0.43 (^{1})	0.72 (*)	0.67 (2.22×10^{-16})
Spirochetes	0.99 (0.04)	0.95 (0.11)	N/A	0.65 (3×10^{-3})

AUC for all 48 components of the leave-phylum-out cross-validation study.

Here (^{1}) means the observed AUC had no significance, (*) indicates a p-value of less than 1×10^{-40}, and "N/A" indicates that there were no test examples for the associated component.

Results: MvirDB

MvirDB is the database of virulence genes compiled at Lawrence Livermore National Labs. We used it as an independent test set for our SVM method; seeing which MvirDB genes that were not part of our initial training set showed up as positive examples according to the SVM. Results are good.

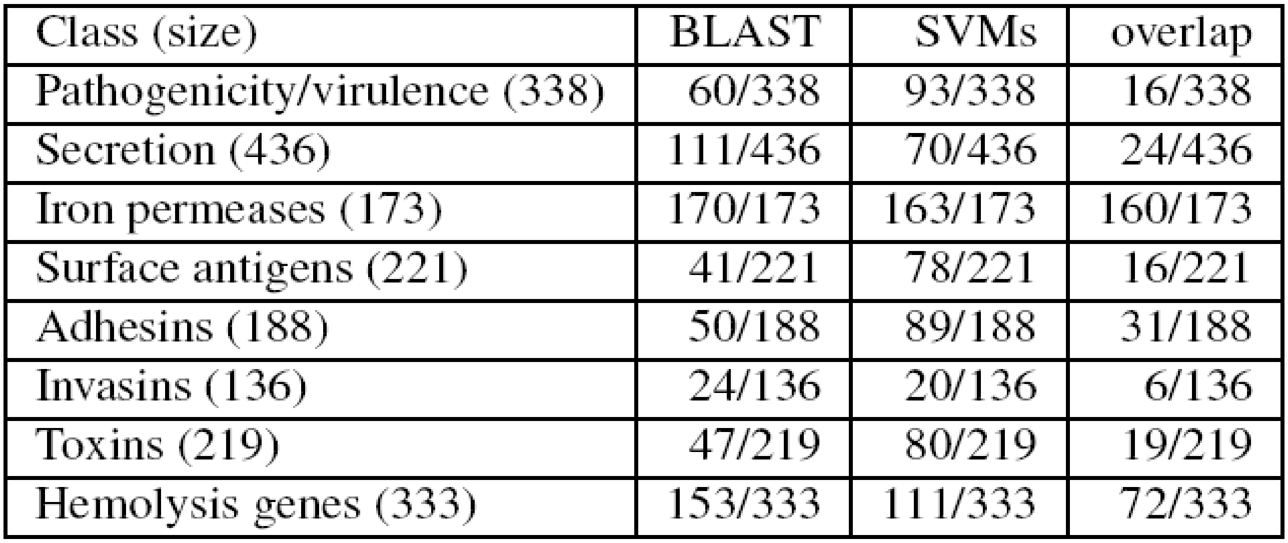


Class	AUC
Pathogenicity/virulence	0.941 (2.822×10^{-3})
Secretion	0.874 (2.133×10^{-1})
Iron Permeases	0.767 (2.523×10^{-1})
Surface antigens	0.942 (1.829×10^{-8})
Adhesins	0.972 (4.091×10^{-5})
Invasins	N/A
Toxins	0.890 ($< 1 \times 10^{-40}$)
Hemolysis genes	0.918 (2.061×10^{-12})

AUC for all 8 components of the MvirDB experiment.

What we get is different from BLAST

It would not be interesting if our SVM was identifying the same set of genes that BLAST would. Fortunately, we get a very different set of genes. Thus the SVM method adds value to any method based on BLAST and identifies a significant number of positive examples BLAST would miss.



Class (size)	BLAST	SVMs	overlap
Pathogenicity/virulence (338)	60/338	93/338	16/338
Secretion (436)	111/436	70/436	24/436
Iron permeases (173)	170/173	163/173	160/173
Surface antigens (221)	41/221	78/221	16/221
Adhesins (188)	50/188	89/188	31/188
Invasins (136)	24/136	20/136	6/136
Toxins (219)	47/219	80/219	19/219
Hemolysis genes (333)	153/333	111/333	72/333

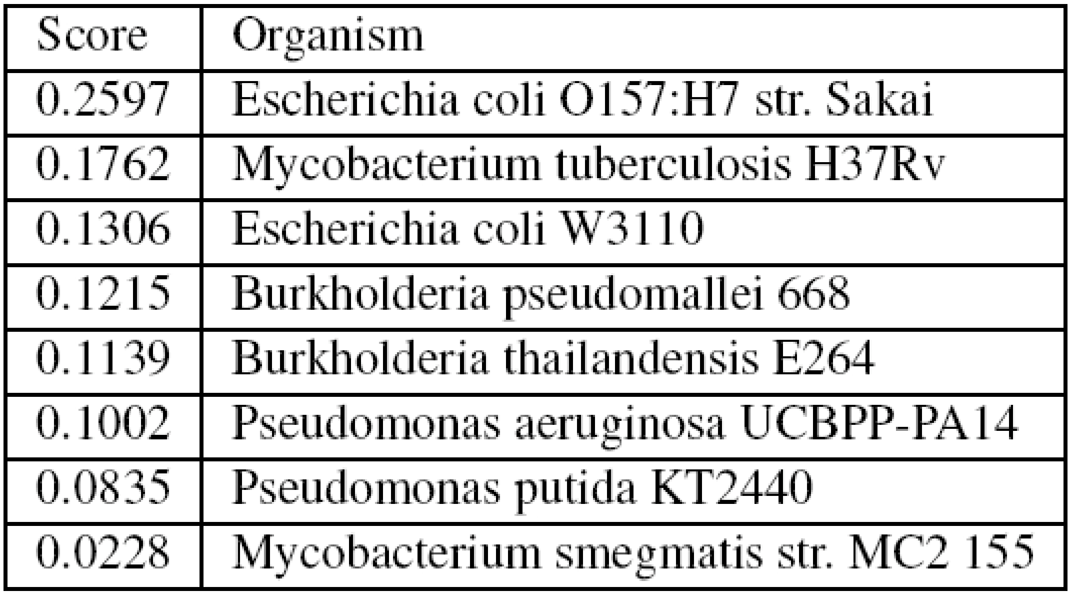
Comparison between BLAST and SVMs.

Results: genome profiling

- By counting the number of genes predicted to be associated with pathogenicity in the top .15% of genomes from unknown organisms, we can score their *relative pathogenicity*.

Furthermore, our SVM method, when run against a whole genome, can generate a global score for how pathogenic a particular bacterial species or strain is. We look at the top .15% scoring genes in the genome, and generate a pathogenicity score to compare globally across genomes.

Conclusion



Score	Organism
0.2597	Escherichia coli O157:H7 str. Sakai
0.1762	Mycobacterium tuberculosis H37Rv
0.1306	Escherichia coli W3110
0.1215	Burkholderia pseudomallei 668
0.1139	Burkholderia thailandensis E264
0.1002	Pseudomonas aeruginosa UCBPP-PA14
0.0835	Pseudomonas putida KT2440
0.0228	Mycobacterium smegmatis str. MC2 155

Conclusion

- There are short signatures of pathogenicity that have functional implications
- These methods are orthogonal to BLAST; produce different results
- Can help with high-throughput annotation of unknown bacterial genomes
- Can help with Environmental Sampling
- A list of currently unannotated genes that our method predicts may have pathogenic function is available on request.

Our next task is to use this to find pathogenic islands.

Detect Inserted Foreign DNA

- Goal: Identify when a string of foreign DNA has been artificially inserted into a host

- Approach: Use methods of unsupervised anomaly detection

- Intuition: Foreign DNA should have anomalous codon bias, compared with host organism
- Methods
  - Distance from Centroid
  - One-class Support Vector Machines
  - Compression-based Methods

- Results: Unsatisfactory. Host organisms themselves contain large amounts of ‘anomalous’ DNA due to horizontal gene transfer

Detecting Phylogenetic Outliers

- Target: a familiar genome with foreign genes inserted

- Rationale: e.g., insertion of pathogenic genes from anthrax in the genome of a common, easily spread bacterium. Also occurs naturally (LGT), but rarely, so would focus attention on a small subset of genes including malicious insertions.

- Traditional approach (e.g., Lerat et al., *PLoS Biol.*, 2003) relies on phylogenetic tree construction, which can be done in many different ways, each with its own limitations.

Our goal is to identify genes whose evolutionary history appears different from the rest of the genes in a genome. This will serve to focus our attention on genes that might have been maliciously inserted from another organism, as well as on genes whose history is different due to natural causes (lateral gene transfer, or LGT). Thus, our algorithm may be of independent interest as a complementary way to detect LGT. In conjunction with our predictor for pathogenicity, this method may identify malicious engineered sequences.

- Idea: if a gene is inserted from a foreign organism, its position in the tree will appear to have moved significantly.
- Our approach: use distance rather than trees to find these outliers
- In this example, pairwise distances from gene 3 in species E to gene 3 in species A-D will all be unusual.
- Tested on *E. coli* genome with simulated horizontal gene transfer (swapping genes in from other proteobacteria).

Often, LGT is currently detected using tree-based methods. The problem is that constructing phylogenetic trees is slow (not suitable for scanning whole genomes, generally) and sometimes incorrect. We can solve our problem using the distance data used for tree construction, but without actually building the trees. This makes our approach faster and avoids some of the errors that tree reconstruction methods can make.

Results

- Overall sensitivity: 46%. Specificity: hard to assess because right answer unknown, but we predict a comparable rate of LGT to previous methods.
- Our accuracy is very high in two crucial cases:
- Species not too close to *E. coli*
- Rapidly-evolving species
- where tree methods fail
- 95% sensitivity finding insertions from species with up to 60% sequence identity of the original genome.

To test our methods, we created a data set where we swapped genes into *E. coli* from related organisms and attempt to detect them using our distance-based approach. Our method is strong at detecting swapped genes in two crucial cases: where the swapped sequences are reasonably distant from the original ones, and when they are swapped in from species that are rapidly evolving – in which case tree construction methods don't do very well. Overall, we are able to detect about 46% of the swapped genes, while predicting 8-10% of the genome as having non-standard evolution. (We can't assess the specificity this way, though, because most of these are probably not false-positives; published estimates of the rate of LGT in bacterial genomes range up to 15%, though we think that's a little high, so we're very happy with 8-10% predicted positives.)

- We succeed especially well when traditional tree-based methods fail.

- Our method is fast enough to be used on entire genomes (unlike tree-based methods).

- Only other genome-wide LGT-detection method (DarkHorse) uses BLAST. Comparison on *E. coli* genome (without inserted outliers):
  - Very different results, but some evidence we are right.
  - We predict 27 outliers in *E. coli* that they missed, including selB, thyA, hscC: literature calls these examples of HGT.

We compare our method to tree-based methods and find that we find complementary things, plus tree methods are too slow to use on whole genomes. The only existing whole-genome method that we've found (for LGT) is DarkHorse, based on BLAST. We again find that our results are complementary. In particular, we predict 27 *E. coli* genes have non-standard evolution that they fail to find. We haven't manually verified all of these yet, but so far we know that at least three of them have independent published evidence for being true examples of LGT.

- Our distance-based method for detecting inserted genes (and LGT) works well in cases where tree- and BLAST- based methods fail.

- Our approach should be used in combination with these other methods to identify potential malicious insertions.

Our approach finds real LGT that other methods miss, and are fast enough to use on whole genomes. They should be used in conjunction with predictors for pathogenicity and other LGT approaches to identify candidate malicious insertions.